ML56-TK Linux 模块驱动

NuMicro® 64/32位系列微控制器范例代码介绍

文件信息

应用简述	本范例代码说明如何在 Linux 核心下导入 ML56-TK 驱动
BSP 版本	Linux-5.4.x
开发平台	NuMaker-Base-MA35D1B1 V2.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1. 概述

本范例代码示范如何将 ML56-TK 核心模块引入 Linux，并透过 “input_test” 测试程序译码输入事件成人类可读取的格式。

为了让使用者能简单的桥接 ML56-TK 至应用程序，ML56-TK 驱动将 ML56-TK 作为输入设备引入 Linux，且将触控键视为是一键盘并回报键值(KEY_A, KEY_B…等)以及按键状态(按下/放开)至系统，用户可透过设备树定义欲回报的键值。而触控条/触控滑轮则视为是一触控屏幕并转换原始数据为 X-Y 的坐标回报至系统，X 值代表百分比、Y 值则用来识别此原始数据是来自触控条或触控滑轮。

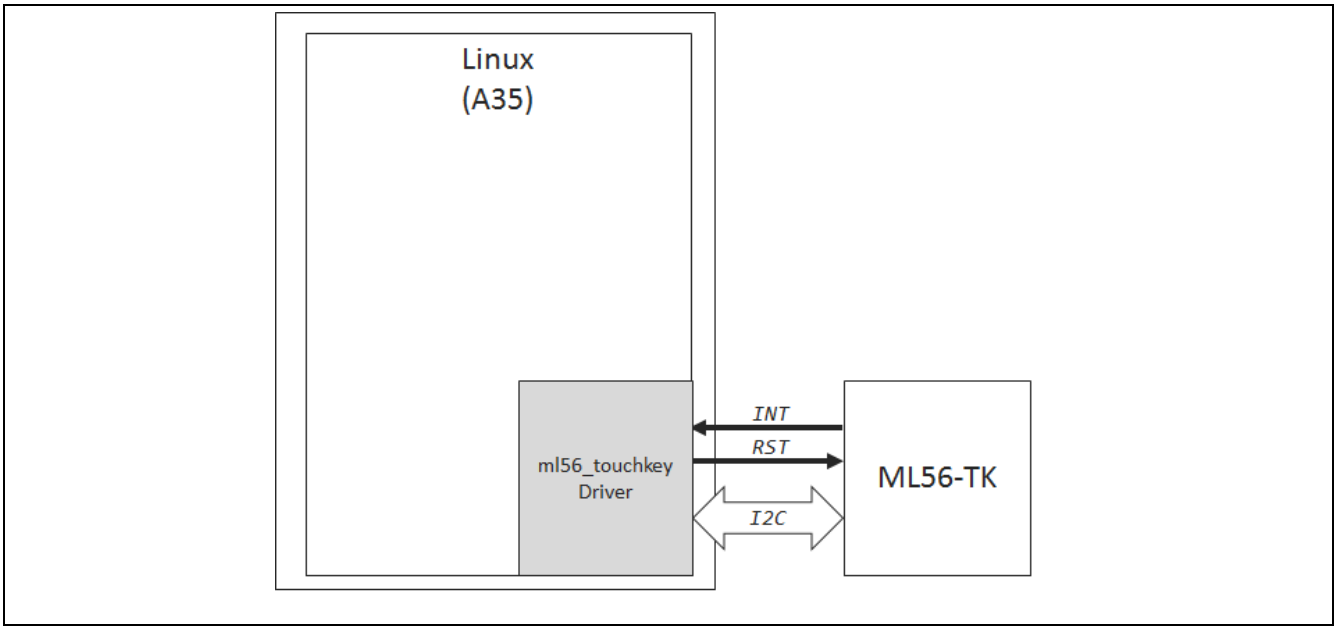

图 1-1 ML56-TK Linux 驱动架构

1.1 驱动设定

ML56-TK 核心模块 I2C 与多功能引脚 Device Tree 的设定参考如下。

1.1.1 I2C 设定

ML56-TK 核心模块由 gpio 驱动 i2c protocol，device tree 设定如下。

```
i2c_gpio0: i2c-gpio-0 {
    reg = <0 0 0 0>;
    compatible = "i2c-gpio";
    gpios = <&gpiom 14 0 /* sda */
      &gpiom 15 0 /* scl */
     >;
    i2c-gpio,delay-us = <20>;        /* ~100 kHz */
    #address-cells = <1>;
    #size-cells = <0>;
    status = "disabled";
};
```

1.1.2 多功能引脚设定

ML56-TK 核心模块多功能引脚，device tree 设定如下。

```
&i2c_gpio0 {
    status = "okay";
   ...
    ml56-tk@52 {
        status = "okay";
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_i2c_gpio0>;
        compatible = "nuvoton,ml56-tk";
        reg = <0x52>;
        reset-gpios = <&gpiom 1 GPIO_ACTIVE_LOW>;
        interrupt-parent = <&gpiok>;
        interrupts = <8 IRQ_TYPE_LEVEL_LOW>;
        irq-gpios = <&gpiok 8 0x00>;
        ml56-tk,keymap = <10 KEY_A>, <11 KEY_B>;
    };
};

/* Add ml56 module pin configuration to device tree's pinctl as below */

&pinctrl {
....
    pcfg_ml56_interrupt: pcfg-pcfg_ml56_interrupt {
     slew-rate = <0>; /* 0:normal, 1:high */
     input-schmitt-disable;  /* input-schmitt-disable, input-schmitt-enable */
     bias-pull-up; /* bias-disable, bias-pull-up, bias-pull-down  */
     power-source = <3300>; /* 1800:1.8v , 3300:3.3v */
     drive-strength = <3>; /* 0 ~ 7 */
    };
```

```
    pcfg_ml56_scl: pcfg-pcfg_ml56_sck {
     slew-rate = <0>;         /* 0:normal, 1:high */
     input-schmitt-disable;   /* input-schmitt-disable, input-schmitt-enable */
     bias-disable;            /* bias-disable, bias-pull-up, bias-pull-down  */
     power-source = <3300>;   /* 1800:1.8v , 3300:3.3v */
     drive-strength = <3>;    /* 0 ~ 7 */
     };

    pcfg_ml56_sda: pcfg-pcfg_ml56_sda {
     slew-rate = <0>;         /* 0:normal, 1:high */
     input-schmitt-disable;   /* input-schmitt-disable, input-schmitt-enable */
     bias-disable;            /* bias-disable, bias-pull-up, bias-pull-down  */
     power-source = <3300>;   /* 1800:1.8v , 3300:3.3v */
     drive-strength = <3>;    /* 0 ~ 7 */
     };

    pcfg_ml56_reset: pcfg-pcfg_ml56_reset {
     slew-rate = <0>;         /* 0:normal, 1:high */
     input-schmitt-disable;   /* input-schmitt-disable, input-schmitt-enable */
     bias-pull-up;            /* bias-disable, bias-pull-up, bias-pull-down  */
     power-source = <3300>;   /* 1800:1.8v , 3300:3.3v */
     drive-strength = <3>;    /* 0 ~ 7 */
    };

   i2c_gpio0 {
       pinctrl_i2c_gpio0: i2c_gpio0grp {
        nuvoton,pins =
        <SYS_GPK_MFPH_PK8MFP_GPIO       &pcfg_ml56_interrupt>,
        <SYS_GPM_MFPL_PM1MFP_GPIO       &pcfg_ml56_reset>,
        <SYS_GPM_MFPH_PM15MFP_GPIO      &pcfg_ml56_scl>,
        <SYS_GPM_MFPH_PM14MFP_GPIO      &pcfg_ml56_sda>;
       };
    };
};
```

1.2 执行结果

本范例代码执行的结果如图 1-2 所示

```
Wed Mar  2 03:50:39 2022.437039 type 0x0001; code 0x0030; value 0x00000001; Key 48 (0x30) press
Wed Mar  2 03:50:39 2022.437039 type 0x0000; code 0x0000; value 0x00000000;
Wed Mar  2 03:50:39 2022.657376 type 0x0001; code 0x0030; value 0x00000000; Key 48 (0x30) release
Wed Mar  2 03:50:39 2022.657376 type 0x0000; code 0x0000; value 0x00000000;
Wed Mar  2 03:50:41 2022.888186 type 0x0001; code 0x001e; value 0x00000001; Key 30 (0x1e) press
Wed Mar  2 03:50:41 2022.888186 type 0x0000; code 0x0000; value 0x00000000;
Wed Mar  2 03:50:42 2022.107135 type 0x0001; code 0x001e; value 0x00000000; Key 30 (0x1e) release
Wed Mar  2 03:50:42 2022.107135 type 0x0000; code 0x0000; value 0x00000000;

Wed Mar  2 03:51:20 2022.610418 type 0x0001; code 0x014a; value 0x00000001; Button 74 press
Wed Mar  2 03:51:20 2022.610418 type 0x0003; code 0x0000; value 0x00000004; Absolute X 4
Wed Mar  2 03:51:20 2022.610418 type 0x0003; code 0x0001; value 0x00000000; Absolute Y 0
Wed Mar  2 03:51:20 2022.807639 type 0x0003; code 0x0000; value 0x00000061; Absolute X 97
```

```
Wed Mar  2 03:51:20 2022.898640 type 0x0001; code 0x014a; value 0x00000000; Button 74 release

Wed Mar  2 03:53:18 2022.847879 type 0x0001; code 0x014a; value 0x00000001; Button 74 press
Wed Mar  2 03:53:18 2022.847879 type 0x0003; code 0x0000; value 0x00000000; Absolute X 0
Wed Mar  2 03:53:18 2022.847879 type 0x0003; code 0x0001; value 0x00000002; Absolute Y 2
Wed Mar  2 03:53:19 2022.544309 type 0x0003; code 0x0000; value 0x00000050; Absolute X 80
Wed Mar  2 03:53:19 2022.644725 type 0x0001; code 0x014a; value 0x00000000; Button 74 release
```

图 1-2 代码执行结果

2. 代码介绍

“input_test” 读取 /dev/input/event* 并且转换事件为人类可读的格式。

```
if (argc > 1) {
     sprintf(name, "/dev/input/event%d", atoi(argv[1]));
     if ((fd = open(name, O_RDWR, 0)) >= 0) {
         printf("%s: open, fd = %d\n", name, fd);
         for (i = 0; i < LED_MAX; i++) {
             event.time.tv_sec  = time(0);
             event.time.tv_usec = 0;
             event.type         = EV_LED;
             event.code         = i;
             event.value        = 0;
             write(fd, &event, sizeof(event));
         }

         while ((rc = read(fd, &event, sizeof(event))) > 0) {
             printf("%-24.24s.%06lu type 0x%04x; code 0x%04x;"
                    " value 0x%08x; ",
                    ctime(&event.time.tv_sec),
                    event.time.tv_usec,
                    event.type, event.code, event.value);
             switch (event.type) {
             case EV_KEY:
                 if (event.code > BTN_MISC) {
                     printf("Button %d %s",
                            event.code & 0xff,
                            event.value ? "press" : "release");
                 } else {
                     printf("Key %d (0x%x) %s",
                            event.code & 0xff,
                            event.code & 0xff,
                            event.value ? "press" : "release");
                 }
                 break;
             case EV_REL:
                 switch (event.code) {
                 case REL_X:      tmp = "X";       break;
                 case REL_Y:      tmp = "Y";       break;
                 case REL_HWHEEL: tmp = "HWHEEL";  break;
                 case REL_DIAL:   tmp = "DIAL";    break;
                 case REL_WHEEL:  tmp = "WHEEL";   break;
                 case REL_MISC:   tmp = "MISC";    break;
                 default:         tmp = "UNKNOWN"; break;
                 }
                 printf("Relative %s %d", tmp, event.value);
                 break;
             case EV_ABS:
                 switch (event.code) {
                 case ABS_X:        tmp = "X";        break;
                 case ABS_Y:        tmp = "Y";        break;
                 case ABS_Z:        tmp = "Z";        break;
```

```
                    case ABS_RX:       tmp = "RX";       break;
                    case ABS_RY:       tmp = "RY";       break;
                    case ABS_RZ:       tmp = "RZ";       break;
                    case ABS_THROTTLE: tmp = "THROTTLE"; break;
                    case ABS_RUDDER:   tmp = "RUDDER";   break;
                    case ABS_WHEEL:    tmp = "WHEEL";    break;
                    case ABS_GAS:      tmp = "GAS";      break;
                    case ABS_BRAKE:    tmp = "BRAKE";    break;
                    case ABS_HAT0X:    tmp = "HAT0X";    break;
                    case ABS_HAT0Y:    tmp = "HAT0Y";    break;
                    case ABS_HAT1X:    tmp = "HAT1X";    break;
                    case ABS_HAT1Y:    tmp = "HAT1Y";    break;
                    case ABS_HAT2X:    tmp = "HAT2X";    break;
                    case ABS_HAT2Y:    tmp = "HAT2Y";    break;
                    case ABS_HAT3X:    tmp = "HAT3X";    break;
                    case ABS_HAT3Y:    tmp = "HAT3Y";    break;
                    case ABS_PRESSURE: tmp = "PRESSURE"; break;
                    case ABS_DISTANCE: tmp = "DISTANCE"; break;
                    case ABS_TILT_X:   tmp = "TILT_X";   break;
                    case ABS_TILT_Y:   tmp = "TILT_Y";   break;
                    case ABS_MISC:     tmp = "MISC";     break;
                    default:           tmp = "UNKNOWN";  break;
                    }
                    printf("Absolute %s %d", tmp, event.value);
                    break;
                case EV_MSC: printf("Misc"); break;
                case EV_LED: printf("Led");  break;
                case EV_SND: printf("Snd");  break;
                case EV_REP: printf("Rep");  break;
                case EV_FF:  printf("FF");   break;
                    break;
                }
                printf("\n");
            }
            printf("rc = %d, (%s)\n", rc, strerror(errno));
            close(fd);
        }
    }
```

3. 软件与硬件需求

3.1 软件需求

- BSP 版本
  - Linux-5.4.x
- 模块名称
  - ml56-touchkey.ko

3.2 硬件需求

- 电路组件
  - NuMaker-HMI-MA35D1 V2.0
  - ML56-TK Module
- 示意图

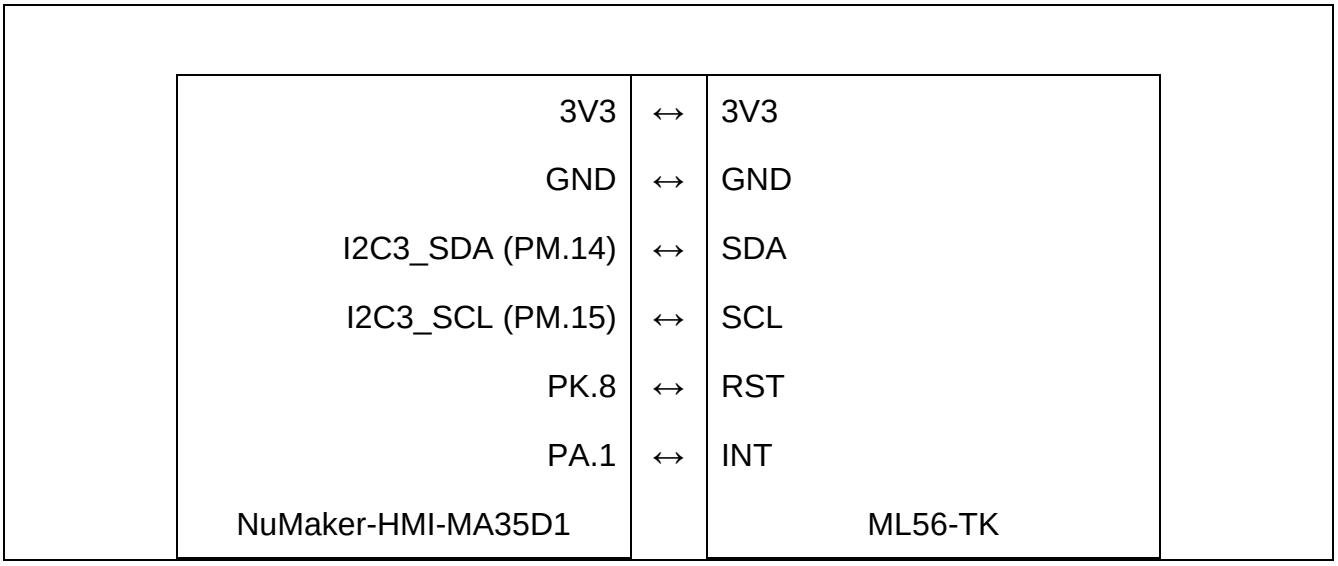

图 3-1 连接示意图

4. 目录信息

EC_MA35D1_ML56-TK_Driver_V1.00	
Src	Source files of ML56-TK device driver
Ko	Kernel object file of ML56-TK device driver
Input_test	The test program of input device
DT_Binding	Declares requirements on the contents of ML56-TK driver

图 4-1 目录信息

5. 范例程序执行

1. 修改所需的引脚(请参考驱动设定)
2. 在编译环境中重新编译核心与模块
3. 将 NuMaker-HMI-MA35D1 开机
4. 复制 ml56-touchkey.ko、input_test 到 NuMaker-HMI-MA35D1 上
5. 导入 ml56-tk 核心模块

```
# insmod ml56-touchkey.ko
  会出现如下画面,红色框部分的数字为应用程序的输入参数
[   61.425658] input: ml56-tk as /devices/platform/0.i2c-gpio-0/i2c-0/0-0052/input/input2
```

6. 执行应用程序

```
# ./input_test 2
```

7. 触碰 ML56-TK 触控键

```
Wed Mar  2 03:50:39 2022.437039 type 0x0001; code 0x0030; value 0x00000001; Key 48 (0x30) press
Wed Mar  2 03:50:39 2022.437039 type 0x0000; code 0x0000; value 0x00000000;
Wed Mar  2 03:50:39 2022.657376 type 0x0001; code 0x0030; value 0x00000000; Key 48 (0x30) release
Wed Mar  2 03:50:39 2022.657376 type 0x0000; code 0x0000; value 0x00000000;
Wed Mar  2 03:50:41 2022.888186 type 0x0001; code 0x001e; value 0x00000001; Key 30 (0x1e) press
Wed Mar  2 03:50:41 2022.888186 type 0x0000; code 0x0000; value 0x00000000;
Wed Mar  2 03:50:42 2022.107135 type 0x0001; code 0x001e; value 0x00000000; Key 30 (0x1e) release
Wed Mar  2 03:50:42 2022.107135 type 0x0000; code 0x0000; value 0x00000000;
```

8. 触碰 ML56-TK 触控条及触控滑轮

```
Wed Mar  2 03:51:20 2022.610418 type 0x0001; code 0x014a; value 0x00000001; Button 74 press
Wed Mar  2 03:51:20 2022.610418 type 0x0003; code 0x0000; value 0x00000004; Absolute X 4
Wed Mar  2 03:51:20 2022.610418 type 0x0003; code 0x0001; value 0x00000000; Absolute Y 0
Wed Mar  2 03:51:20 2022.807639 type 0x0003; code 0x0000; value 0x00000061; Absolute X 97
Wed Mar  2 03:51:20 2022.898640 type 0x0001; code 0x014a; value 0x00000000; Button 74 release
```

6. 修订纪录

Date	Revision	Description
2022.09.27	1.00	初始发布。

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.